

Appl. No. 10/034,220
Amdt. dated February 11, 2005
Reply to Office action of December 9, 2004

Amendments to the Specification:

Please replace paragraph [0004] with the following amended paragraph:

[0004] The heart of any computer system is the motherboard. The motherboard generally contains a microprocessor, main memory array, and various bridge devices which enable hardware and software components to communicate and perform their respective functions. Given the fact that there are many motherboard manufacturers, which may include any of an array of possible microprocessors, there is likewise an array of varying steps and procedures required for software to communicate with hardware. However, software applications, for example word processors and internet browsers, are typically written for a class of computers without distinction as to a specific brand of computer system or the specific type of motherboard. Thus, some functions which software applications perform are abstracted (or taken away) from [I] the software applications such that the specifics for each hardware implementation need not be embodied in the user-level software. The first level of abstraction between user-level programs and hardware typically takes the form of driver programs. Thus, rather than user-level software attempting to communicate directly with hardware, this software need only be programmed to communicate with a respective driver program. The driver program then is assigned the task of communicating with the hardware devices. Performing the task of communicating with hardware may take the form of calling basic input/output system (BIOS) routines to perform very specific tasks. While many BIOS routines are standard across all types of computing systems, most ROM BIOS manufacturers allow original equipment manufacturers (OEMs), companies who make the computers that consumers purchase, to define their own BIOS routines. The problem, however, is keeping track of the many BIOS routines that an OEM may provide.

Appl. No. 10/034,220
Amdt. dated February 11, 2005
Reply to Office action of December 9, 2004

Please replace paragraph [0007] with the following amended paragraph:

[0007] A second method for determining whether a particular BIOS routine is supported is to have the kernel level software (driver) ~~who~~ which does the BIOS call first check a BIOS information table to obtain a BIOS version number, as well as the BIOS date, and compare the information to determine whether the BIOS supports the BIOS routine desired. This, however, cannot guarantee that in every circumstance a correct conclusion regarding support for a particular BIOS routine, and further may require a substantial amount of information to be hard coded in each driver for the comparison step, which significantly increases the footprint of the driver in main memory.

Please replace paragraph [0027] with the following amended paragraph:

[0027] In the preferred embodiment, each BIOS routine is assigned a unique identification number, different than the routine's interrupt category and services number, which calling programs, preferably kernel mode drivers, use to identify the desired BIOS routine. Preferably, the unique identifier for each BIOS routine is of a sufficient number of bits that, for practical purposes, no duplicate identification numbers should exist, even between different OEMs. Stated otherwise, the identification numbers are preferably globally unique. In the preferred embodiments, the identification numbers are Globally Unique Identifiers (GUIDs) generated using the program GUIDGEN.EXE, provided with Microsoft developers tool kit. The GUIDs generated with the GUIDGEN.EXE program are 128 bit identification numbers which are based on the network address of the executing computer, the date and time of running the GUIDGEN.EXE program, along with a random number from a random number generator. ~~One of ordinary skill in the art is aware of the GUIDGEN.EXE program for use in assigning unique identifiers for software programs outside the context of ROM BIOS routines. Thus, one of ordinary skill in the art, now understanding the preferred embodiment of using the GUIDGEN.EXE program in the context of the BIOS routines, could easily understand their application in the BIOS routine context.~~

Appl. No. 10/034,220
Amdt. dated February 11, 2005
Reply to Office action of December 9, 2004

Please replace paragraph [0030] with the following amended paragraph:

[0030] A portion of the BIOS code known as a dispatcher 50 preferably analyses-analyzes the service number passed in the SAL_PROC parameters (or if implemented in an pre-Itanium™ system, the service number stored in the AH register after issuance of the interrupt) and executes the BIOS routine identified by the service number. In the exemplary case of service number 2, the software that performs the desired functions, exemplified by box 52, is called by the dispatcher. After execution of the desired BIOS routine, the dispatcher returns, with or without a return parameter based on the specific type of BIOS routine called.

Please replace paragraph [0031] with the following amended paragraph:

[0031] Although there may be many ways for a driver program to ascertain the precise location of the table which correlates the GUIDs to the service numbers, in the preferred embodiments the driver programs access the industry standard System Management BIOS (SMBIOS) table to ascertain the location of the data table. ~~One of ordinary skill in the art is well aware of how to locate and access information in the SMBIOS table.~~ Preferably, the SMBIOS table has an entry which gives a pointer to the location of the data table that correlates the GUIDs to the service numbers. While it is preferred that the driver programs access the SMBIOS table to make a determination as to the location of the GUID to service number correlation table, it is also within the contemplation of this invention that the driver programs may also know the location of the correlation table directly.